Spectral Measures of Distortion for Change Detection in Dynamic Graphs

Complex Networks 2018, dec 14th, Cambridge (UK)

Luca Castelli Aleardi
Ecole Polytechnique

Maks Ovsjanikov
Ecole Polytechnique

Semih Salihoglu
Univ. of Waterloo

Gurprit Singh
Max Planck Institute

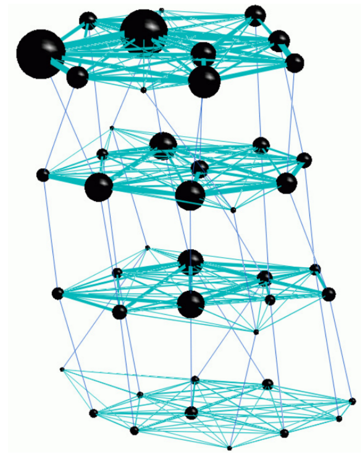
ÉCOLE POLYTECHNIQUE

Visualization of (Dynamic) Networks

("as I have known it")

Network visualization

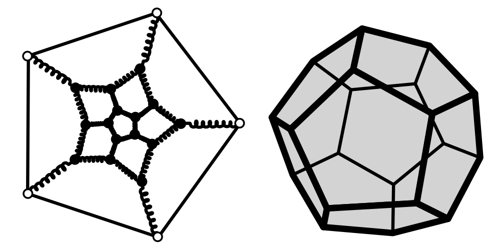
Visualization of temporal networks with time slices (Erten et al. 2004)



[W. Tutte'63] Tutte barycentric embedding

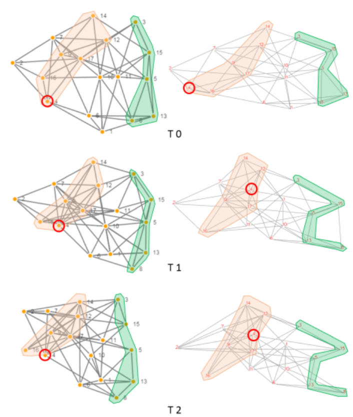
minimize the spring energy

$$E(\rho) := \sum_{(i,j)\in E} |\mathbf{x}(v_i)-\mathbf{x}(v_j)|^2 = \sum_{(i,j)\in E} (x_i-x_j)^2+(y_i-y_j)^2$$



Laplacian-based visualization (Che et al., 2015)

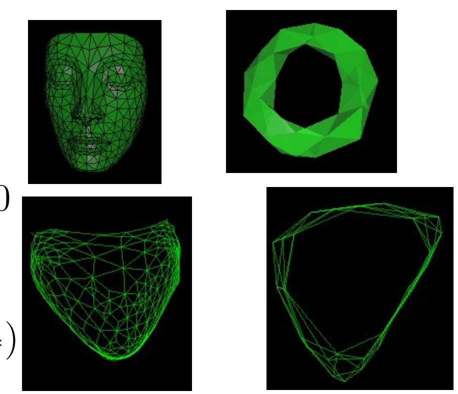
$$E=\sum_{(j,k)\in E_i}\frac{(\|x_j-x_k\|-d_{j,k})^2}{(d_{j,k})^2}+\alpha\sum_{j,k\in V_{i-1}\cap V_i}\frac{(\|x_j-x_k\|-d_{j,k})^2}{(d_{j,k})^2}$$



Spectral layouts **minimize the energy with constraints**

$$\begin{cases} \min_{\underline{x}} E(\underline{x}) := x\, L_G\, x \\ \text{constraint: } \underline{x}^T \cdot \underline{x} = 1 \\ x_M = \sum_i x_i = 0 \qquad \underline{x}^T \cdot 1_n = 0 \end{cases}$$

$$(x_1,\ldots,x_d) = \left(\frac{v_2[i]}{\sqrt{\lambda_2}}, \frac{v_3[i]}{\sqrt{\lambda_3}}, \ldots, \frac{v_{d+1}[i]}{\sqrt{\lambda_{d+1}}}\right)$$



Incremental layout method (Crnovrsanin et al., 2015)

Dynamic spectral layout (Brandes et al., 2007)

...

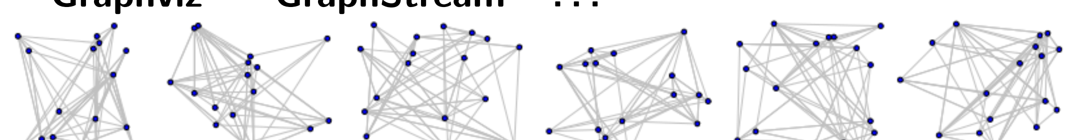
... **(many others)**

...

Many tools for dealing with dynamic networks

Gephi **SoNIA** **Cytoscape**

Graphviz **GraphStream** **...**



Layout of the newcomb dataset produced by SoNIA software

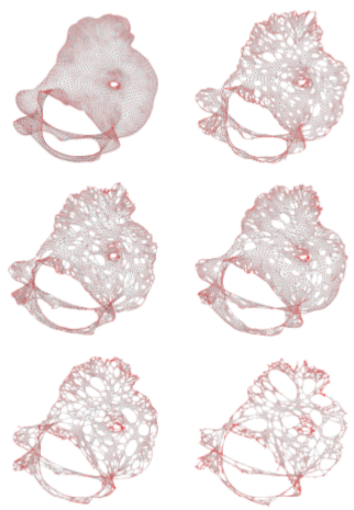
Dynamic network visualization: existing works

General purpose node-link layouts (online setting)

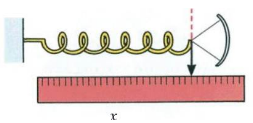
Problem compute a new layout for G_i given the layouts of $(G_1, G_2, \dots, G_{i-1})$

Constraints **readibility, mental map preservation**

Node pinning
(Frishman Tal, 2008)



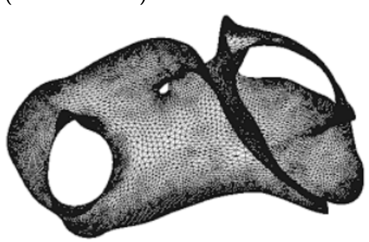
Spring embedder (Eades, 1984)
(Fruchterman and Reingold, 1991)

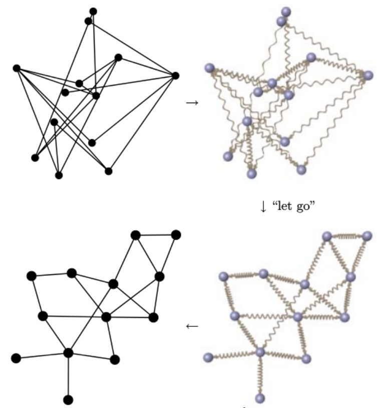


$$\mathbf{F}_a(v) = c_1 \cdot \sum_{(u,v)\in E} \log(dist(u,v)/c_2)$$

$$\mathbf{F}_r(v) = c_3 \cdot \sum_{u \in V} \frac{1}{\sqrt{dist(u,v)}}$$

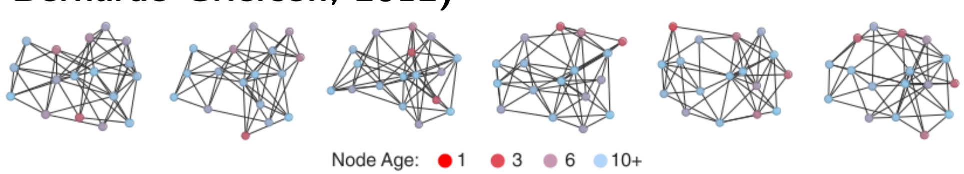
(4elt network)





images from Kaufman Wagner (Springer, 2001)

Vertex ages (Gorochowski Di Bernardo Grierson, 2012)



Newcomb fraternity dataset

Dynamic network visualization: existing works

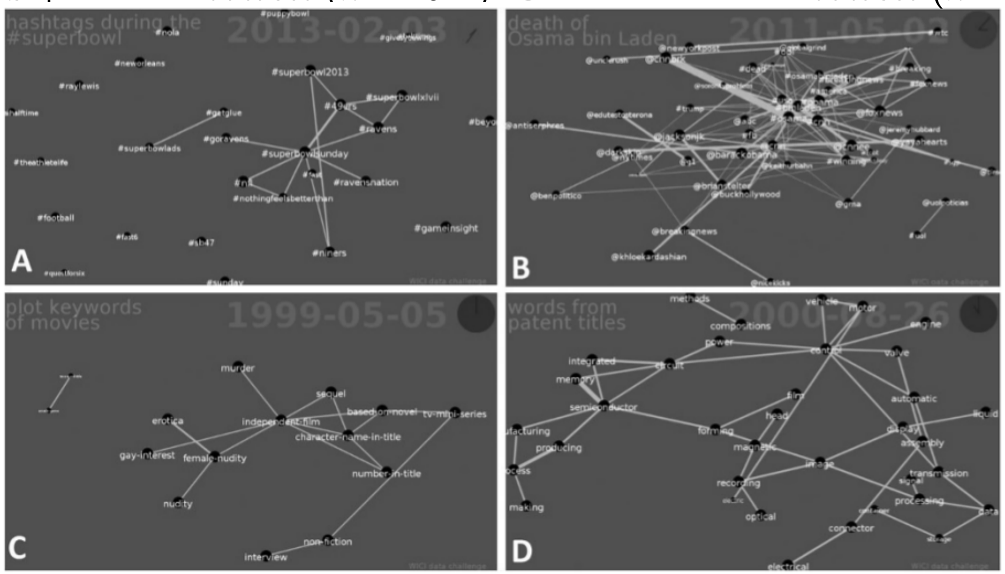
General purpose node-link layouts (online setting)

Problem capture persistent trends in network evolution of the sequence $(G_1, G_2, \ldots, G_i)$

Goal **smooth transition between consecutive snapshots**

Fast filtering and animation of dynamic networks (Grabowicz, Aiello, Menczer, 2014)

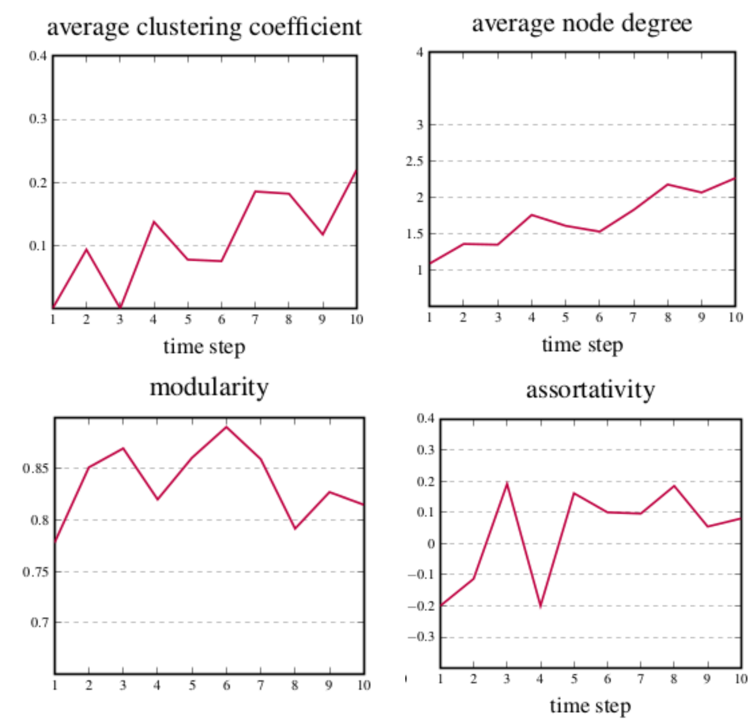
Super Bowl dataset $(n = 49K)$ Osama Bin Laden dataset $(n = 95K)$



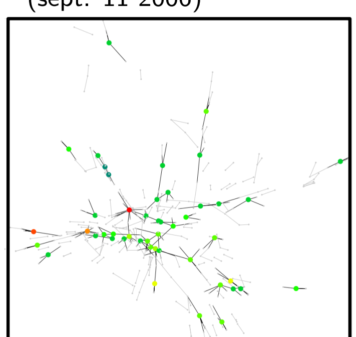
IMBD keywords $(n = 1K)$ US patents dataset $(n = 414K)$

Threshold filtering (keep 10% of nodes)

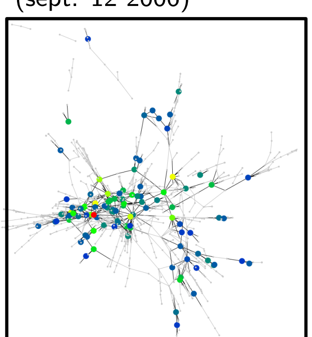
Facebook growth over ten days (sep. 5-15 2006)



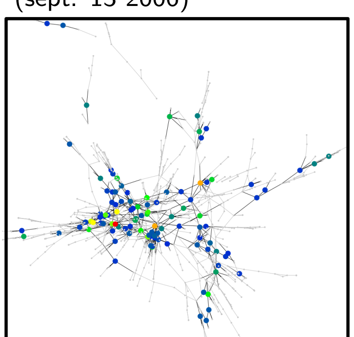
(sept. 11 2006)



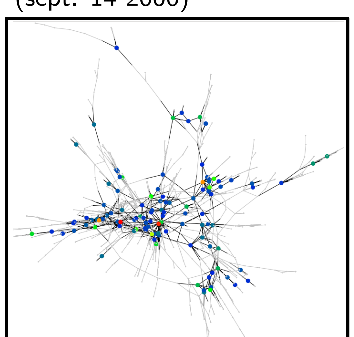
(sept. 12 2006)



(sept. 13 2006)



(sept. 14 2006)



Dynamic network visualization: online setting

Input a sequence $S = \{G_1, G_2, \ldots, G_T\}$

Assumptions $G_i = (V_i, E_i)$ general (simple) undirected, unweighted
the vertex set is constant: there is bijective correspondence between V_{i-1} and V_i

---

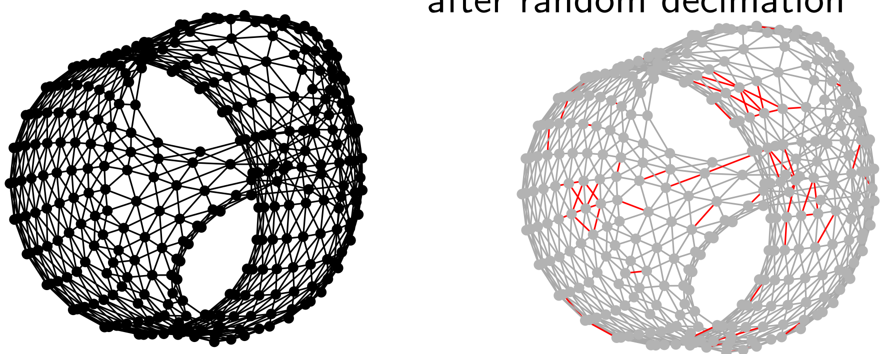
Problem: detect and highlight relevant changes at time i from G_{i-1} to G_i

arbitrary Network evolution edge deletions/additions, vertex deletions/additions

---

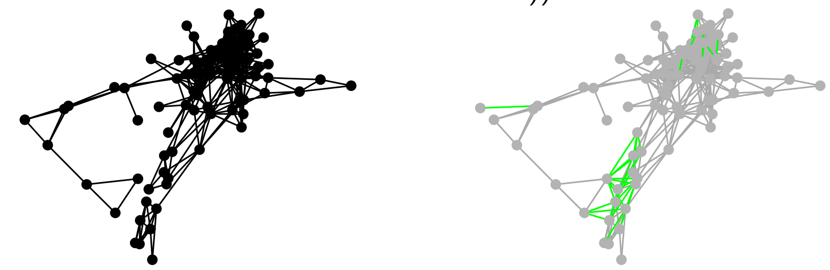
tested datasets networks from real-world applications, social networks, proximity graphs, ...

after random decimation



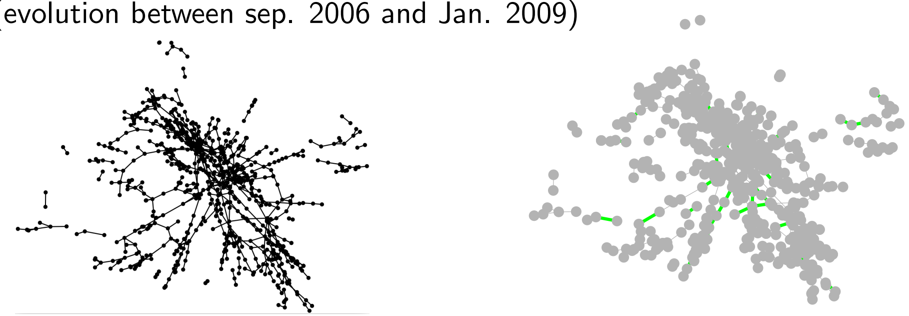
dwt307 network
(from the SuiteSparse Matrix Collection)

SG dataset (Dynamics of person-to-person interactions, Cattuto et al. 2010))

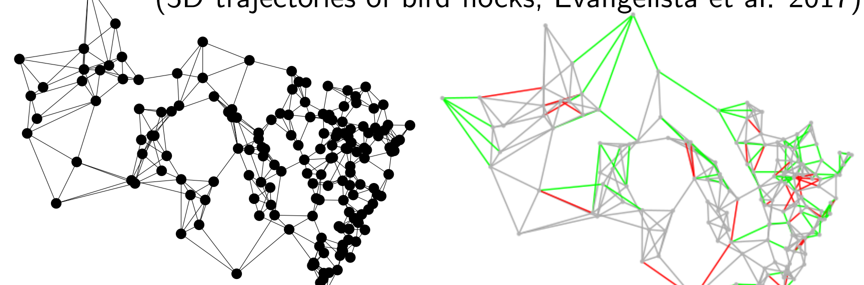


face-to-face proximity interactions between individuals visiting the Science Gallery of Dublin

social network growth (Facebook dataset)
www.eecs.wsu.edu/~yyao/StreamingGraphs.html
(evolution between sep. 2006 and Jan. 2009)



bird flocking dataset (proximity graph)
(3D trajectories of bird flocks, Evangelista et al. 2017)

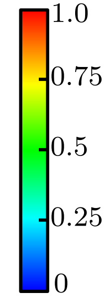


(220 edge modifications in a single timestep)

Detection of relevant changes, via a distortion function

Graph distortion: natural requirements

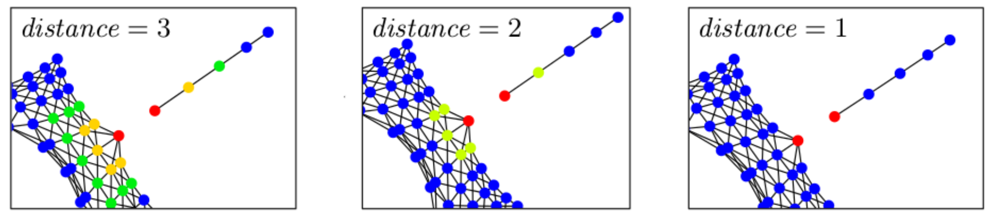
Main problem: detect most relevant changes from G_i to G_{i+1}



Approach: find a distortion function $f : V \to [0, 1]$ assigning high values to vertices with most relevant changes

Desirable requirements:

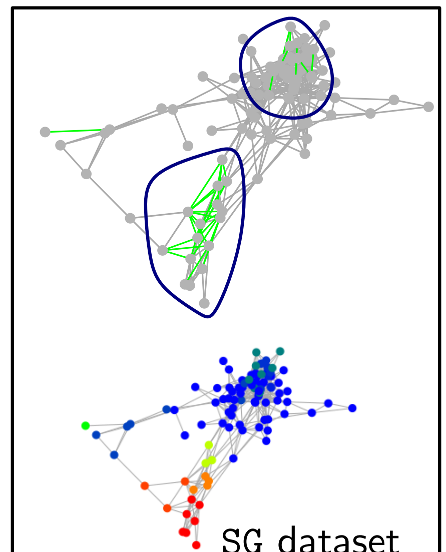
distance-to-modification effect

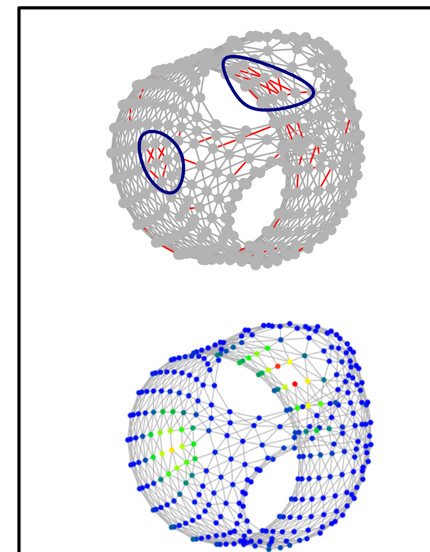


highlight regions that changed most

detect high degree vertices and clusters

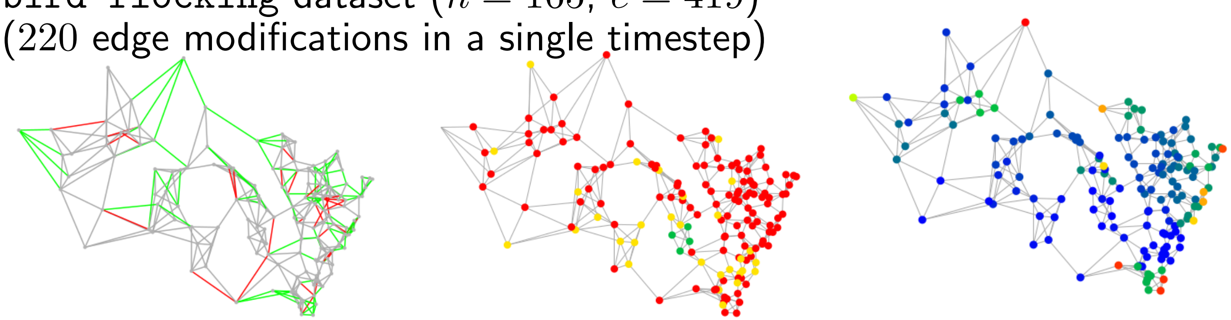
smooth and multi-scale behaviour





smooth behaviour even in the case of dramatic changes

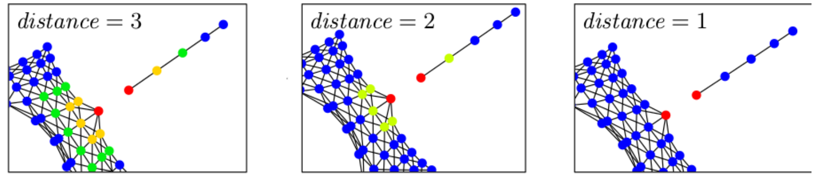
`bird flocking` dataset ($n = 165$, $e = 419$)
(220 edge modifications in a single timestep)

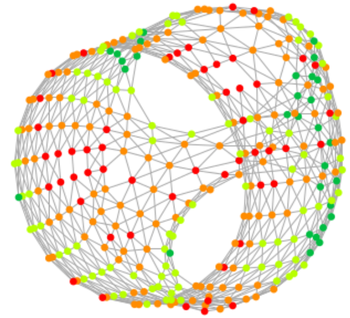


Graph distortion: related works

distance-to-modification distortion **(related to node pinning weights, Frishman and Tal, 2008)**

$$\delta_{DM}(u) := 1 - (\alpha^{1-\frac{dist(u)}{\beta}})$$





constants α, β are input parameters

$dist(u) :=$ distance from the closest modification

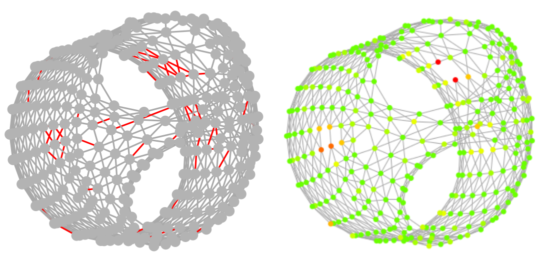
no detection of high degree vertices/clusters
limited smoothness

vertex age distortion **(related to vertex age, Gorochowski et al. '12)**

$$\delta_{VA}(u) := e^{-\beta age(u,i)}$$

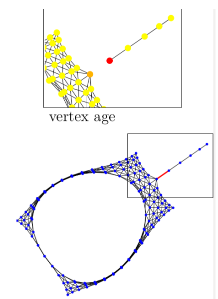
constant β is an input parameter

$age(u,i)$ is the age of vertex u at timestep i



(exponential decay, with respect to the age)

$$age(u,i+1) = \begin{cases} age(u,i)+1, & \text{if } u \text{ has degree 0 in } G_{i+1} \\ age(u,i)\frac{\mathscr{A}^{rem}_{i+1}(u)}{\mathscr{A}^{tot}_{i+1}(u)}+1, & \text{otherwise} \end{cases}$$

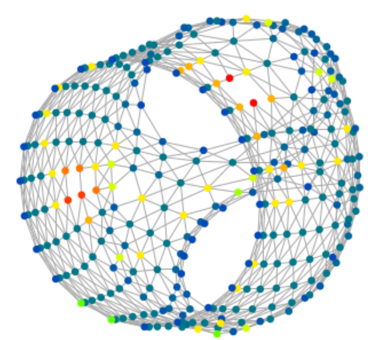


$$\mathcal{A}^{tot}_i(u) = \mathcal{A}^{rem}_i(u) + \mathcal{A}^{add}_i(u) + \mathcal{A}^{del}_i(u)$$

vertex strength distortion **(related to vertex strength, Grabowicz, Aiello, Menczer, 2014)**

$strength(u,i)$ is the strength of vertex u at timestep i

- $strength(u,1) := degree(u)$ in G_1
- for each edge (u,v) added/remoevd: update the strength of u and v (by an addictive factor)
- the strength is decreased periodically, by a multiplicative factor $c_f \leq 1$ ($c_f = 0.25$ for our experiments)



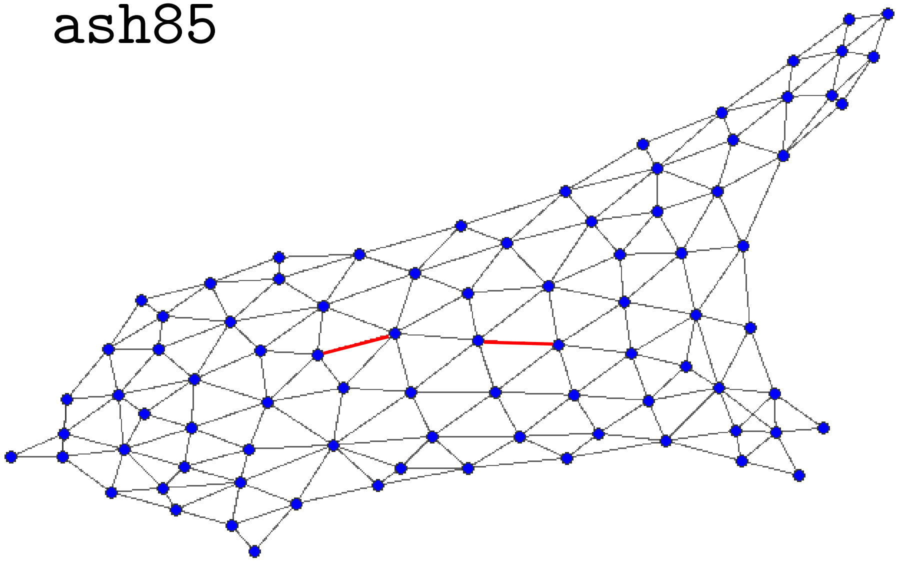
Our contribution: spectral measures of graph distortion

Spectral distortion: intuition

Main idea: find f optimizing an energy $E(G_i, G_{i+1}, f)$

example $E_{\text{vertex ratio}}(G_i, G_{i+1} f) = \frac{\sum_u f(u)^2 d_{i+1}(u)}{\sum_u f(u)^2 d_i(u)}$ $d_i(u) := degree(u)$ in G_i

`ash85`



Spectral distortion: intuition

minimizing

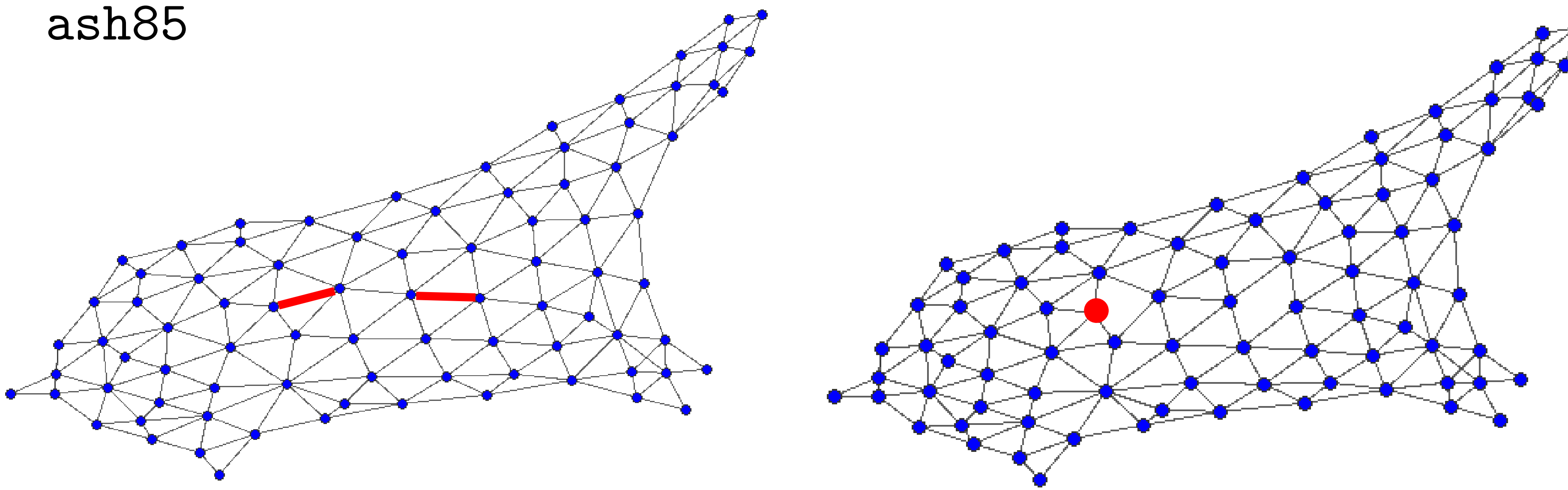
Main idea: find f optimizing an energy $E(G_i, G_{i+1}, f)$

example $E_{\text{vertex ratio}}(G_i, G_{i+1} f) = \frac{\sum_u f(u)^2 d_{i+1}(u)}{\sum_u f(u)^2 d_i(u)}$ $d_i(u) := degree(u)$ in G_i

drawback the solution is the indicator function

$$f(u) = \begin{cases} 1 & u \text{ s.t. } \frac{d_{i+1}(u)}{d_i(u)} \text{ is minimal} \\ 0 & \text{otherwise} \end{cases}$$

ash85



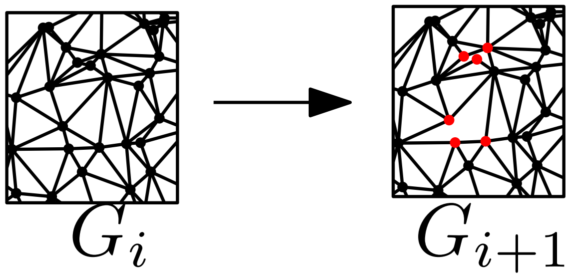
Spectral distortion: our pipeline

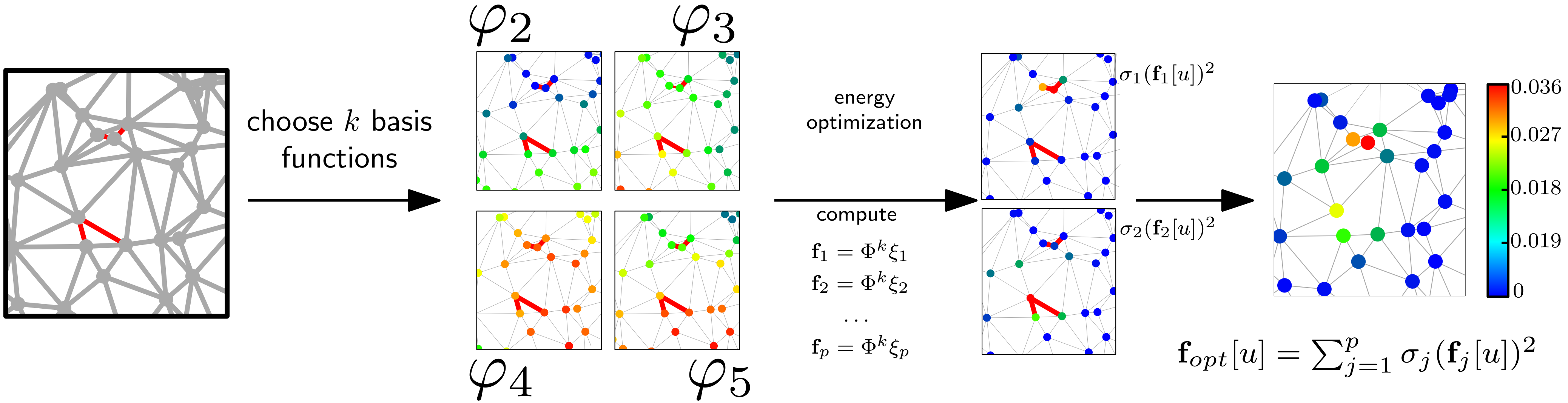
Main idea: find $f : V \to [0,1]$ optimizing an energy $E(G_i, G_{i+1}, f)$

the distortion function is of the form: $f = \sum_{j \leq k} a_j \varphi_j$

where $\varphi_i : V \to R$

f is in the linear sub-space spanned by k functions $\{\varphi_j\}$





Option 1: $$E_{\text{vertex diff}}(G_i, G_{i+1}, f) = \frac{\sum_u f(u)^2 (d_i(u) - d_{i+1}(u))^2}{\sum_u f(u)^2 d_i(u)}$$

Option 2 $$E_{\text{edge diff}}(G_i, G_{i+1}, f) = \frac{\sum_{u,v} (f(u) - f(v))^2 (w_i(u,v) - w_{i+1}(u,v))^2}{\sum_u f(u)^2 d_i(u)}$$

Option 3 $$E_{\text{vertex ratio}}(G_i, G_{i+1} f) = \frac{\sum_u f(u)^2 d_{i+1}(u)}{\sum_u f(u)^2 d_i(u)}$$

Option 4 $\dots$

Notation:
$d_i(u) := degree(u),\ u \in V(G_i)$
$w_i(u,v) := weight(u,v),\ (u,v) \in E(G_i)$

Spectral distortion: computation

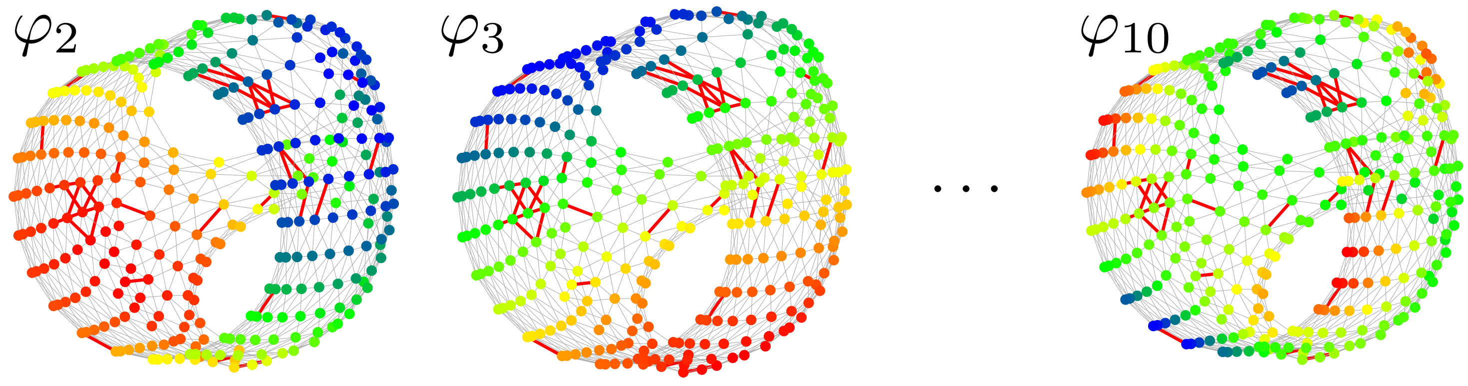
General pipeline:

Step 0: set user parameters

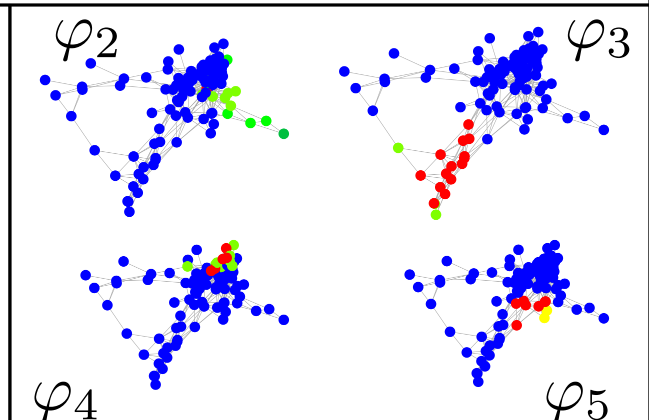
- choose the scale parameter k (usually $k << n$)
- choose region parameter p **(related to the number of regions to detect)**

Step 1: choose a well adapted family of k basis functions $\{\varphi_1, \ldots, \varphi_k\}$

Option 1: eigenfunctions



Option 2: clustering basis functions



Step 2: choose a well adapted distortion energy $E(G_i, G_{i+1}, f)$

For example (when only edge additions or removals can occur)

$$E_{\text{vertex diff}}(G_i, G_{i+1}, f) = \frac{\sum_u f(u)^2 (d_i(u) - d_{i+1}(u))^2}{\sum_u f(u)^2 d_i(u)}$$

Spectral distortion: computation

General pipeline:

Step 1:

Option 1: eigenfunctions

solve the eigenproblem

$$L_{G_i}\varphi = \lambda D_{G_i}\varphi$$

takes the smallest k eigenvalues

Option 2: compute a partition into k clusters **(we use the Louvain algorithm)**

given the partition $V(G_i) = \{R_i, R_2, \dots, R_k\}$

$$\varphi_j(u) = \begin{cases} \frac{1}{1+dist(u)} & u \in R_j \\ 0 & \text{otherwise} \end{cases}$$

Step 2: optimize energy $E(G_i, G_{i+1}, f)$

let $\Phi^k := [\varphi_1, \dots, \varphi_k]$

solve the generalized eigenproblem

$$\boxed{S_{i+1}\xi = \sigma S_i \xi}$$

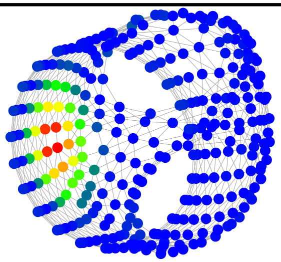
$$S_{i+1} = (\Phi^k)^T L^-_{G_i,G_{i+1}} \Phi^k$$

$$S_i = (\Phi^k)^T D_{G_i} \Phi^k = Id$$

$(\sigma_1, \dots, \sigma_k)$ eigenvalues $(\xi_1, \dots, \xi_k)$ eigenvectors

the optimal function f is of the form: $\mathbf{f}_{opt} = \Phi^k \xi_{max}$

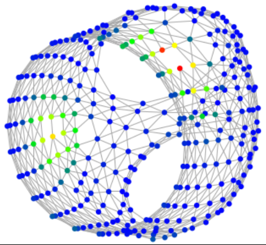
Step 3: get the distortion

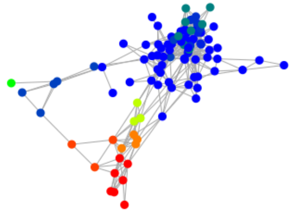


define the spectral distortion: $\delta(u) := \frac{(\mathbf{f}_{opt}[u])^2}{\max_v (\mathbf{f}_{opt}[v])^2}$

or better, take p eigenvectors: (where $\mathbf{f}_j = \Phi^k \xi_j$)

$$\boxed{\delta(u) := \sum_{j \le p} \sigma_j (\mathbf{f}_j[u])^2}$$





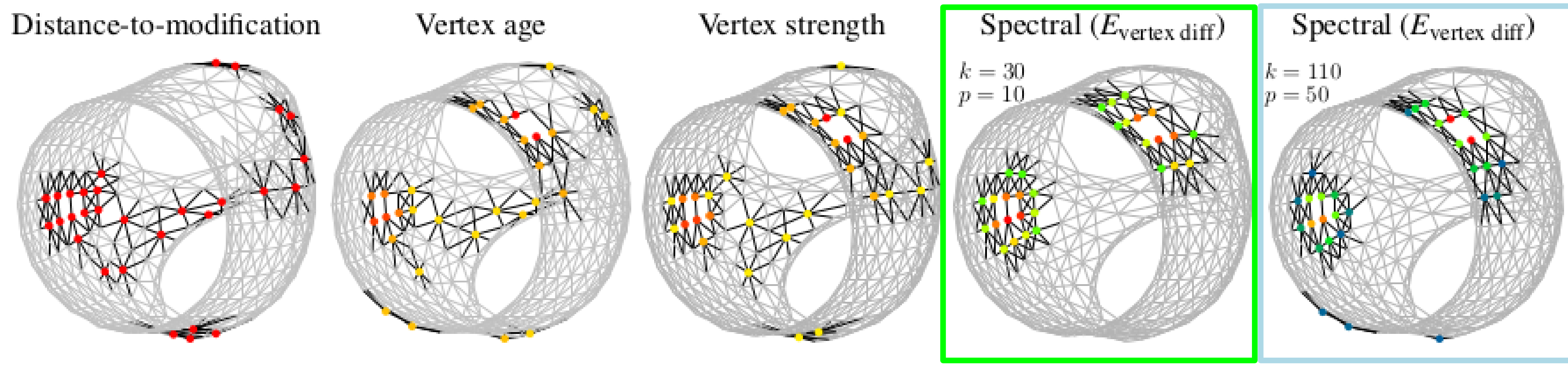
Experimental evaluation

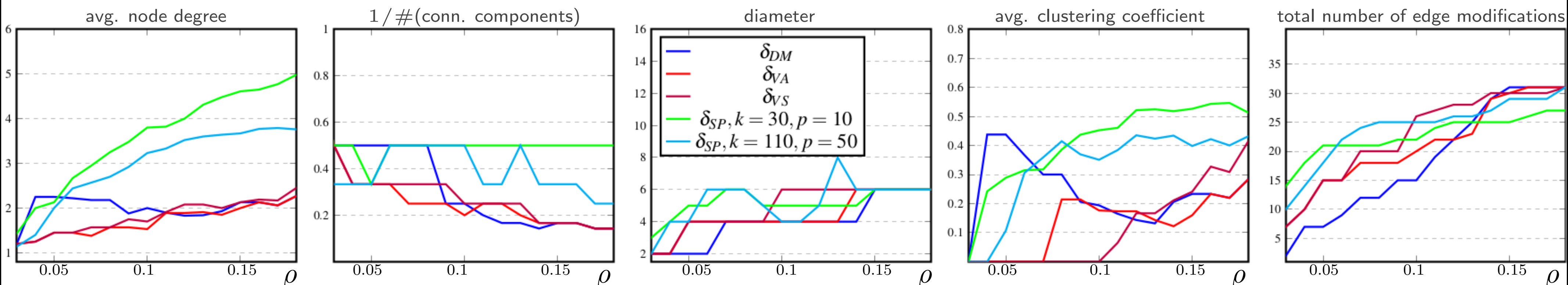
Quantitative evaluation

Threshold filtering approach: keep a fraction $\varrho \in [0,1]$ of the nodes with highest distortion

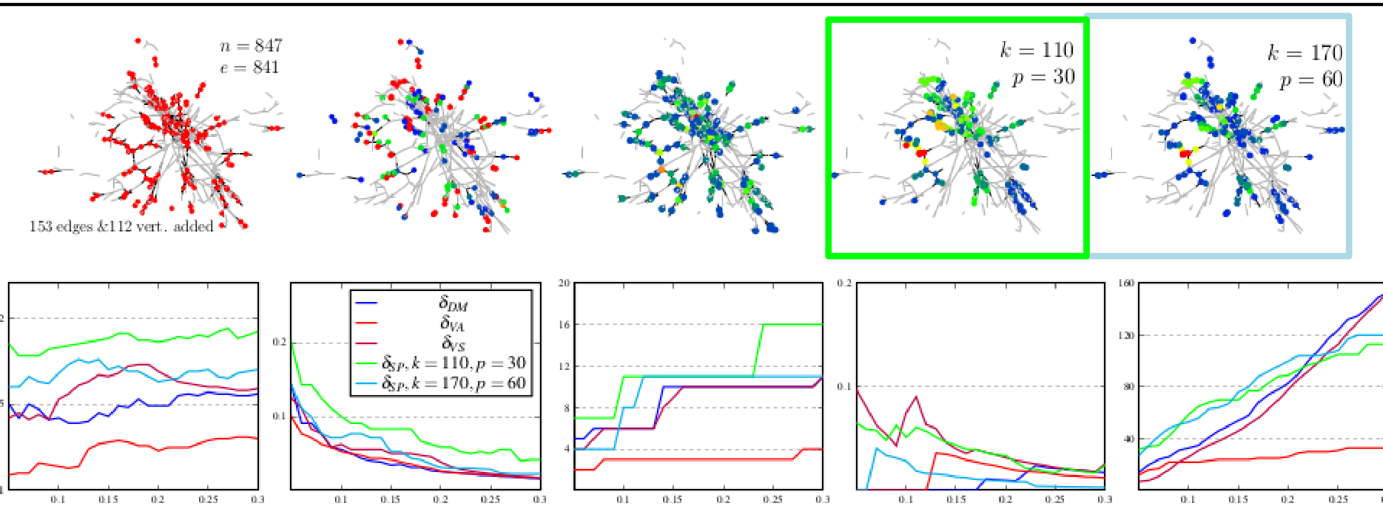
Comparison: evaluate the evolution of structural parameters, between G_i and G_{i+1}

dwt307network, $\varrho = 0.1$ (after decimation)





Facebook-growth, $\varrho = 0.1$



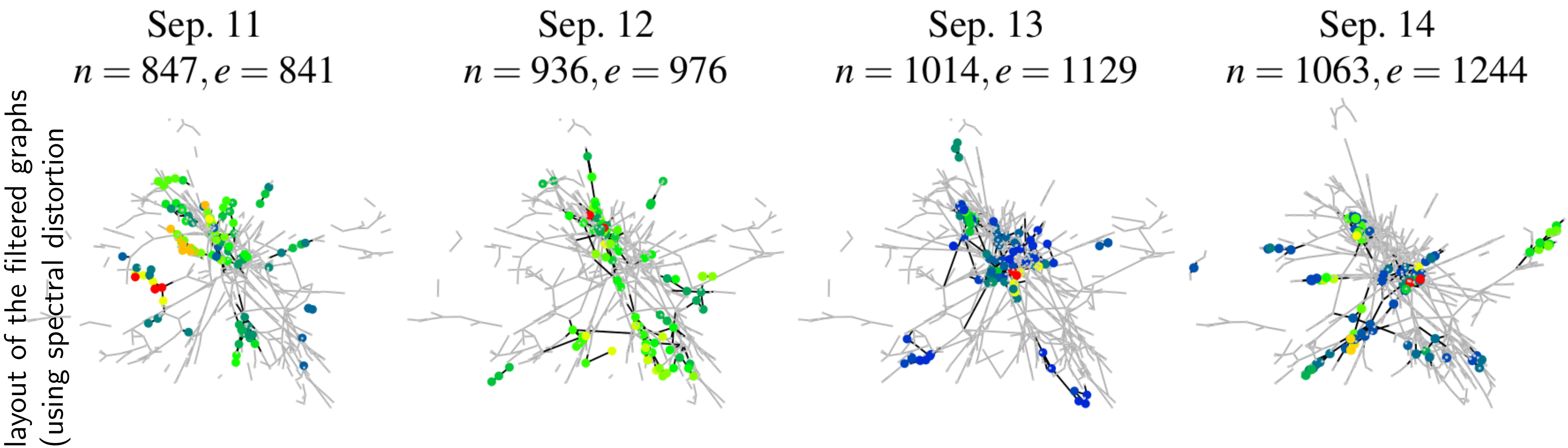
Advantages of our spectral distortion:

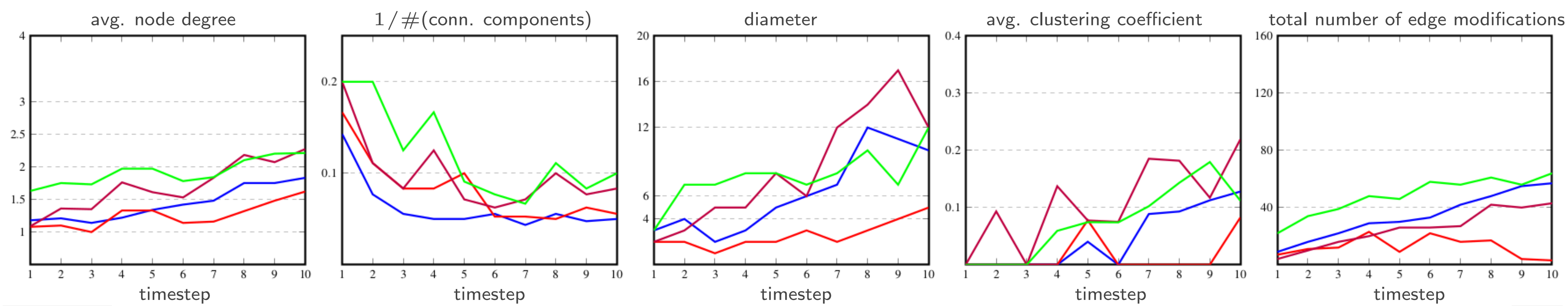
- detect more local changes (for small values of ϱ)
- detected regions are more dense and less cluttered
- less drastic fluctuations of structural properties
- better distinguish relevant changes from local noise

Quantitative comparison

Threshold filtering approach: keep a fraction $\varrho \in [0, 1]$ of the nodes with highest distortion

Comparison: for a fixed value of ϱ evaluate the evolution of structural parameters over the full sequence $\{G_1, G_2, \ldots G_T\}$





`Facebook growth` over ten days (sep. 5-15 2006)

results are obtainedkeeping 10% of nodes with highest distortion ($\varrho = 0.1$)

Time performances

Time performances comparison

We evaluate the computation time (results are expressed in seconds)

			Distortion vertex age	Spectral distortion Step 1 ($k=40$)		Spectral distortion Steps 2 and 3	Force-directed layout one iteration	
Network	vertices	edges		eigen-basis	region functions	$k=40$	FR91	with octrees
SG	96	399	0.0003	0.037	0.002	0.0003	0.002	0.001
dwt307	307	1.1K	0.0015	0.048	0.004	0.0007	0.009	0.003
3elt	4720	13.7K	0.009	0.22	0.075	0.003	0.73	0.11
barth5	15606	61.4K	0.026	0.84	0.32	0.011	15.2	0.57

let us assume
$n = |V_i| \approx |V_{i+1}|$

Distortion vertex age: $O(|E_i|)$

Eigen-basis: $\approx O(n \cdot k^2)$

Region functions: $\approx O(n \log n)$

Steps 2 and 3: $O(k^3 + kn + pn) = O(n)$

FR91: $O(|E_i| + n^2)$

With octrees: $O(|E_i| + n \log n)$

$(k << n)$
$(p \leq k)$

Thank you for your attention